

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): Vugranam C. Sreedhar
Docket No.: YOR920010262US2
Serial No.: 09/925,580
Filing Date: August 9, 2001
Group: 2191
Examiner: Ted T. Vo

Title: Method and Apparatus for Programming Software Components

REPLY BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Appellant hereby replies to the Examiner's Answer, mailed September 11, 2007 (referred to hereinafter as "the Examiner's Answer"), in an Appeal of the non-final rejection of claims 2-4, 10-12, and 18 in the above-identified patent application.

REAL PARTY IN INTEREST

A statement identifying the real party in interest is contained in Appellant's Appeal Brief.

RELATED APPEALS AND INTERFERENCES

A statement identifying related appeals is contained in Appellant's Appeal Brief.

STATUS OF CLAIMS

A statement identifying the status of the claims is contained in Appellant's Appeal Brief.

5

STATUS OF AMENDMENTS

A statement identifying the status of the amendments is contained in Appellant's Appeal Brief.

SUMMARY OF CLAIMED SUBJECT MATTER

10

A Summary of the Invention is contained in Appellant's Appeal Brief.

STATEMENT OF GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 2-4, 10-12, and 18 are rejected under 35 U.S.C. §102(b) as being anticipated by Lavender.

15

CLAIMS APPEALED

A copy of the appealed claims is contained in an Appendix of Appellant's Appeal Brief.

20

ARGUMENT

Independent Claims 4, 12 and 18

25

Independent claims 4, 12, and 18 were rejected under 35 U.S.C. §102(b) as being anticipated by Lavender. Regarding claim 4, the Examiner asserts that Lavendar discloses utilizing an attach command to attach at least one of said at least one input port to a class. In the Examiner's Answer (top of page 5), the Examiner asserts that "the attachment is performed by 'attach()' or the instruction 'bool attached.'" In the Response to Appellant's arguments section of the Examiner's Answer, the Examiner asserts that the instruction attach()...or the programming instruction "bool attached"...is the same as the "attach command" in the Appellant's program.

30

Appellant notes that the “attach” in Lavendar *attaches an output port of one module to an input port of another module*. (See, top of page 6.) Lavendar does *not* utilize an attach command to *attach an input port to a class*.

Regarding the Examiner’s assertion that “the attachment is performed by ‘attach()’ or the instruction ‘bool attached,’” Appellant notes that a statement such as “attach()” that attaches *an input port to a class* cannot be found in the cited statements, and that Lavendar teaches that the “bool attached” statement is a test that is “true if a module is attached.” (See, comments for the cited statement on the middle of page 6.) Therefore, contrary to the Examiner’s assertion, the statement “bool attached” disclosed in Lavendar does *not* attach an input port to a class. Thus, the statement “bool attached” (as disclosed in Lavendar) is *not* the same as the attach command of the present invention which attaches an input port to a class (e.g., *attach xin BoolClass*, described below).

As previously noted, the present invention *utilizes an explicit attach command to attach an input port to a class*. For example, the present disclosure teaches that “a component *must attach each of its input ports* to a concrete class within it.” (Page 3, lines 3-4; emphasis added.) The present disclosure also teaches that “each input port 112 in a component 110 should be attached (*using the attach command*) to some concrete class with in the component 110.” (Page 12, lines 5-6; emphasis added.) Finally, the present disclosure teaches that

the following code segment defines a component 110, referred to as BooleanComp, implementing the template, BooleanTempl, defined above:

```
component BooleanComp implements BooleanTempl {
  attach xin BoolClass ; // attach input port xin to class
  BoolClass ;
  BooleanComp() {...} // constructors.
  class BoolClass implements Bool {
    boolean not(boolean x) { ...} ;
    boolean nand (boolean x, boolean y) {...}
  }
}
```

Thus, the input port xin is “attached” to the class BoolClass.

(Page 6, line 23, to page 7, line 6; emphasis added.)

Independent claims 4, 12, and 18 require utilizing an attach command to attach at least one of said at least one input port to a class

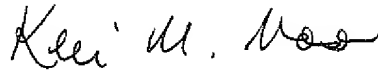
Thus, Lavender does not disclose or suggest utilizing an attach command to attach at least one of said at least one input port to a class, as required by independent
5 claims 4, 12, and 18

Conclusion

The rejections of the cited claims under section 102 in view of Lavender et al. are therefore believed to be improper and should be withdrawn. The remaining
10 rejected dependent claims are believed allowable for at least the reasons identified above with respect to the independent claims

The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully,



Date: November 12, 2007

Kevin M. Mason
Attorney for Applicant(s)
Reg. No. 36,597
Ryan, Mason & Lewis, LLP
1300 Post Road, Suite 205
Fairfield, CT 06824
(203) 255-6560

APPENDIX

1. (Cancelled)

5 2. The method of claim 4, further comprising the step of
allowing said software component to access an external environment only through said
output port.

 3. The method of claim 4, further comprising the step of
10 allowing a client to access said software component only through said input port.

 4. A method executed by a processor for programming a
software component, said method comprising the steps of:

 defining properties of said software component, including at least one
15 input port and at least one output port;

 providing a software mechanism for instantiating said software
component; and

 utilizing an attach command to attach at least one of said at least one input
port to a class.

20

5. (Cancelled)

6. (Cancelled)

25 7. (Cancelled)

8. (Cancelled)

9. (Cancelled)

30

10. The system of claim 12, wherein said processor is further configured to allow said software component to access an external environment only through said output port.

5 11. The system of claim 12, wherein said processor is further configured to allow a client to access said software component only through said input port.

12. A system for programming a software component, said
10 system comprising:

a memory that stores computer-readable code; and

a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:

15 define properties of said software component, including at least one input port and at least one output port;

provide a software mechanism for instantiating said software component;

and

utilize an attach command to attach at least one of said at least one input port to a class.

20

13. (Cancelled)

14. (Cancelled)

25 15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

30

18. An article of manufacture for programming a software component, said system comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

5 a step to define properties of said software component, including at least one input port and at least one output port;

a step to provide a software mechanism for instantiating said software component; and

10 a step to utilize an attach command to attach at least one of said at least one input port to a class.

19. (Cancelled)

20. (Cancelled)

15

EVIDENCE APPENDIX

There is no evidence submitted pursuant to § 1.130, 1.131, or 1.132 or entered by the Examiner and relied upon by appellant.

RELATED PROCEEDINGS APPENDIX

There are no known decisions rendered by a court or the Board in any proceeding identified pursuant to paragraph (c)(1)(ii) of 37 CFR 41.37.